Move a shape to the center of the scene

```
(ns unwavering-briars
  (:require
   [opencv3.utils :as u]
   [opencv3.colors.html :refer :all]
   [opencv3.core :refer :all]))
```

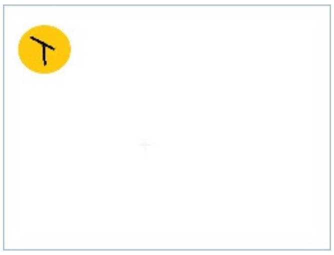
```
nil
```

The plan is defined as follow:

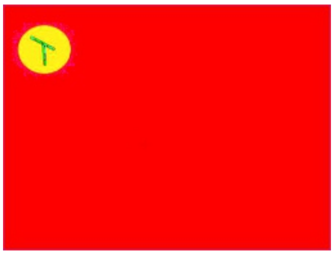
- load the picture, and add a quick border to get it to show properly here.
- go into HSV color space to be able to select only the object we want use a color range
- create a mask

```
(def img
  (-> "resources/morph/cjy6M.jpg"
      imread
      (u/resize-by 0.4)
      (copy-make-border! 1 1 1 1
        BORDER_CONSTANT
        (->scalar "#aabbcc"))))

(u/mat-view img)
```



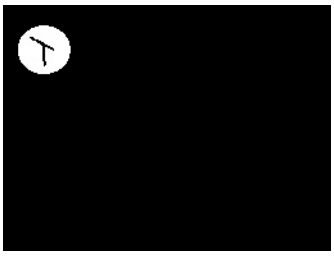
```
(def hsv
  (-> img clone (cvt-color! COLOR_BGR2HSV)))
(u/mat-view hsv)
```



thresh hold on yellow in hsv space

```
(def mask-on-yellow (new-mat))
(in-range hsv (new-scalar 20 100 100) (new-scalar 40 255 255) mask-on-yellow)

(u/mat-view mask-on-yellow)
```



find contours, and select first 1, (actually only one)

```
(def contours (new-arraylist))
(find-contours mask-on-yellow contours (new-mat) RETR_EXTERNAL CHAIN_APPROX_SIMPLE)
(count contours)
```

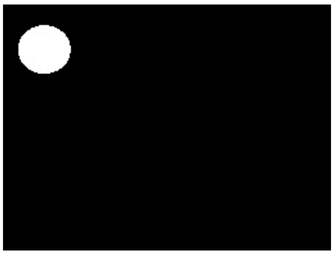
```
1
```

```
(def background-color  (->scalar "#000000"))
; mask type CV_8UC1 is important !!
(def mask (new-mat (rows img) (cols img) CV_8UC1 background-color))

(u/mat-view mask)
```

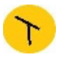
Here we draw the one contour we have found, using the *FILLED* setting of draw-contours.

```
(draw-contours mask contours 0 (new-scalar 255.0) FILLED)
(u/mat-view mask)
```



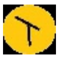
```
(def box
  (bounding-rect (first contours)))
(def item
  (submat img box))

(u/mat-view item)
```



```
(def segmented-item
  (new-mat (rows item) (cols item) CV_8UC3 background-color))
(copy-to item segmented-item (submat mask box) )

(u/mat-view segmented-item)
```



```
(def center
  (new-point (/ (.cols img ) 2 ) (/ (.rows img) 2)))

(def center-box
  (new-rect
    (- (.-x center ) (/ (.-width box) 2))
    (- (.-y center ) (/ (.-height box) 2))
    (.-width box)
    (.-height box)))

(def result (u/mat-from img))
(set-to result (->scalar "#000000"))
```

```
#object[org.opencv.core.Mat 0x2e48b2d3 "Mat [ 192*256*CV_8UC3, isCont=true, isSubmat=false,
nativeObj=0x7fecf585f7a0, dataAddr=0x158eba040 ]"]
```

```
(def final (submat result center-box))
(copy-to segmented-item final (new-mat))

(u/mat-view result)
```